NGS Based Haplotype Assembly Using Matrix Completion

1 ABSTRACT

We use matrix completion methods for haplotype assembly from NGS reads to develop the new HapSVT, HapNuc, and HapOPT algorithms. This is performed by applying a mathematical model to convert the reads to an incomplete matrix and estimating unknown components. This is followed by quantizing and decoding the completed matrix in order to generate haplotypes. These algorithms are compared to the recently addressed SDhaP algorithm for the real fosmid data. It is shown that reconstruction rate and the MSE of these algorithms outperform the SDhaP. Also, the MEC score of the HapOPT is lower than that of the SDhaP with almost the same running time.

Keyword : Haplotype assembly, Single nucleotide polymorphism, Computational biology,Minimum error correction, Matrix completion, Singular value decomposition.

2 INTRODUCTION

The Single Nucleotide Polymorphism (SNP) is a kind of genetic variation with a frequency greater than 1% in population. In diploid organisms, genomes are organized into pairs of chromosomes, a paternal and a maternal copy. The sequence of SNPs on each copy of a pair of chromosomes is called a haplotype. A genotype is the conflation of two haplotypes on the homologous chromosomes. An SNP is called homozygous, if a pair of alleles at this locus is made up of two identical nucleotides, and is heterozygous, otherwise. From the evolutionary point of view, the SNP has been occurred as a consequence of mutation. Since the rate of mutation is low, twice mutation of a specific locus is too rare. Thus, it is usual to assume that the majority of SNPs are bi-allelic which means that each SNP can be chosen from just two of the four possible nucleotides, *i.e.*, A, T, C, and G [1]. Here, we similarly use the same assumption. The haplotype is widely used in the genome wide association studies, clinical genetics, linkage analysis, drug-design, and personalized medicine [2].

To extract a haplotype, one may use the following three approaches where the last two ones are computational based:

1) Applying high-cost experimental and expensive methods for every single individual which is not desirable [2].

2) Haplotype phasing wherein the haplotypes are inferred from the genotypes of multiple individuals. As such, a method based on the maximum parsimony assumption [3] and statistical methods like the SHAPEIT, developed based on the Hidden Markov Model (HMM) [1, 4] may be mentioned.

Note that using this approach, the haplotype of an individual can not be found separately and also is challenged by low-frequency variants and *de novo* variants [2].

3) Estimating haplotypes from Next Generation Sequencing (NGS) reads *i.e.* the nucleotide sequence of fragments. Using this approach, mentioned as the haplotype assembly, haplotyping of a single individual becomes feasible. In this regard, HapCUT [5], HapTree [6], and HapSAT [7] are three famous methods which are developed based on probabilistic models. These methods are sensitive to the selected model and thus fragile to the model error. A recent method for haplotype assembly is the SDhaP [8] which has shown accurate results compared to the HapCut, HapTree, and ReFHap [9]. This heuristic method which makes use of correlation clustering and non-convex optimization does not guarantee reaching the global optimum.

The innovation of this article is threefold. First, in Section 3 matrix completion is shown for haplotype assembly application in a mathematical form. Secondly, we propose three new algorithms including Haplotype assembly based on Singular Value Thresholding (HapSVT), Haplotype assembly based on Nuclear norm minimization (HapNuc), and Haplotype assembly based on OPTSPACE (HapOPT) in Section 4.

In Section 5, these algorithms are evaluated using computer simulations and compared to the recent method, SDhaP, in terms of reconstruction rate, mean square error and minimum error correction score. Section 6 concludes the paper.

3 MODEL OF HAPLOTYPES

To exploit NGS reads as the raw data, a computational modeling is needed. To do so, first we convert the sequence of nucleotides which can be either reads or haplotypes into a sequence of numbers. The SNP nucleotides are converted to 1 and -1 for the wild and rare alleles, respectively [10]. As an example, Table 1 depicts the alleles of the β_2AR gene [3] for which the maternal and paternal haplotypes of an individual are shown by $\boldsymbol{h}_m$ and $\boldsymbol{h}_p$, respectively. The corresponding codewords based on the above modeling are presented in the last column.

Tab. 1: Haplotpyes of β_2AR genes and its corresponding code word.

	Nucleotides										Code word
Allels	G/A	C/A	G/A	C/G	T/C	T/C	T/C	G/A	C/G	G/A	{1/-1,1/-1,...}
$\boldsymbol{h}_m$	A	C	G	G	C	C	C	G	G	G	{-1,1, 1,-1,-1,-1,-1, 1,-1,1}
$\boldsymbol{h}_p$	G	C	A	C	T	T	T	A	C	G	{ 1,1,-1, 1, 1, 1, 1,-1, 1,1}

It is assumed that each read has been aligned to the reference genome. Then, the non-SNP sites of each read are omitted, and thus, the i-th read with the length of l_i has just the information of l_i sites from the whole l sites of the haplotype. Then, the i-th read is coded using the procedure described in Table 1 and is completed by adding zeros for the length of l as shown for 10 aligned reads in Table 2. For example, for the 1st row, we get {-1 1 1 0 0 0 0 0 0 0} which consists of 3 sites of ± 1 and 7 sites of zeros.

Without loss of generality, by showing each row by the vector $\boldsymbol{r}_i, i = 1, ..., N$, we can generate the read matrix $\boldsymbol{R}$ where N is the number of reads as shown below. $\boldsymbol{R}$ is an incomplete version of $\boldsymbol{H}$ which consists of the maternal and paternal haplotypes as its rows, and thus, its rank is

Tab. 2: Example of aligned reads for β_2AR genes and the considered code words.

Reads	Nucleotides										Code words									
1	A	C	G								-1	1	1	0	0	0	0	0	0	0
2			G	G	C	C					0	0	1	-1	-1	-1	0	0	0	0
3			G	G					G	G	0	0	1	-1	0	0	0	0	-1	1
4	G	C	A	C	T	T					1	1	-1	1	1	1	0	0	0	0
5			A	C			T	A	C	G	0	0	-1	0	0	1	1	-1	1	1
6	G	C			T	T					1	1	0	0	1	1	0	0	0	0
7		C			C					G	-1	1	0	0	-1	0	0	0	0	1
8	A	C			C	C	C				-1	1	0	0	-1	-1	-1	0	0	0
9	G			C			T	A	C		1	0	0	1	0	0	1	-1	1	0
10			A	C					C	G	0	0	-1	1	0	0	0	0	1	1

2. Having obtained a low rank matrix, we may utilize matrix completion approach to solve the problem. Specifically, by estimating the zero entries of $\boldsymbol{R}$, it becomes complete as shown by the matrix $\boldsymbol{H}$ which has the same dimension as $\boldsymbol{R}$, *i.e.*, $N \times l$ where l denotes the haplotype length.

$$\boldsymbol{R} = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{1}$$

$$\boldsymbol{H} = \begin{bmatrix} -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \end{bmatrix} \tag{2}$$

According to the definition of $\boldsymbol{H}$, only two of its rows are different and thus the desired haplotypes are given by

$$\boldsymbol{h}_m = \begin{bmatrix} -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix}, \tag{3}$$

$$\boldsymbol{h}_p = \begin{bmatrix} 1 & 1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 \end{bmatrix}. \tag{4}$$

Then, these vectors can be decoded to the sequence of nucleotides using the first row of Table 1. To the best of our knowledge, no algorithm has been reported to distinguish between the maternal and paternal haplotypes and therefore $\boldsymbol{h}_p$ and $\boldsymbol{h}_m$ may be interchanged with each other.

4 PROPOSED METHODS

Here, we present three algorithms for haplotype assembly whose general block diagram is illustrated in Figure 1. The goal is to estimate $\boldsymbol{h}_p$ and $\boldsymbol{h}_m$ as the output. The first two parts of the block diagram, *i.e.*, converting nucleotides to sequences of numbers and preparing a read matrix were explained in Section 3.

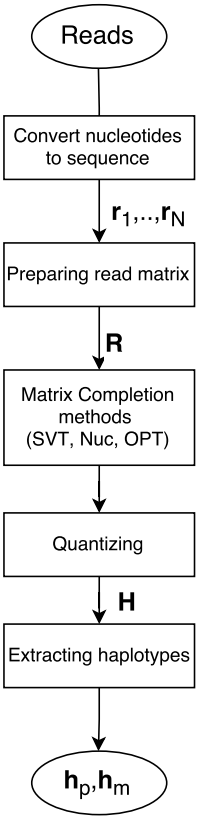

Fig. 1: Block diagram of the proposed algorithms.

We consider an incomplete matrix $\boldsymbol{R}$ with a few known entries where the set of indices of known entries is given by Ω. Then, we intend to estimate the unknown entries based on rank assumption. Mathematically, this is defined by the following optimization problem:

$$\text{Find } \boldsymbol{H} \qquad \text{subject to } \operatorname{rank}(\boldsymbol{H}) = r,\ \boldsymbol{H}_{ij} = \boldsymbol{R}_{ij} \text{ for } (i,j) \in \Omega. \tag{5}$$

Assuming $r = 2$, (1) can be converted to [11]

$$\min_{\boldsymbol{H}} \operatorname{rank}(\boldsymbol{H}) \qquad \text{subject to } \boldsymbol{H}_{ij} = \boldsymbol{R}_{ij} \text{ for } (i,j) \in \Omega. \tag{6}$$

To solve (2), three approaches named as nuclear norm minimization, Singular Value Thresholding (SVT), and OPTSPACE have already been reported [12]. Based on these algorithms, in the following we introduce three novel algorithms for the purpose of haplotype assembly mentioned as the HapSVT, HapNuc, and HapOPT.

4.1 Haplotype assembly using HapSVT

To explain the proposed HapSVT algorithm, we first introduce the SVT which is based on Singular Value Decomposition (SVD) [13] defined for the read matrix $\boldsymbol{R}$ as

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H, \quad \boldsymbol{\Sigma} = \text{diag}(\sigma_i) \quad i = 1, ..., r \tag{7}$$

where H denotes the hermitian operator, and $\boldsymbol{U}$ and $\boldsymbol{V}$ have orthonormal columns with the dimension of $N \times r$ and $l \times r$, respectively. By applying the soft-thresholding operator, or namely, the singular value shrinkage operator $D_\tau(\cdot)$ to $\boldsymbol{H}$, we obtain

$$D_\tau(\boldsymbol{H}) = \boldsymbol{U} D_\tau(\boldsymbol{\Sigma}) \boldsymbol{V}^H \tag{8}$$

where

$$D_\tau(\boldsymbol{\Sigma}) = \text{diag}\big(\max\{\sigma_i - \tau, 0\}\big). \tag{9}$$

It is worth noting that $D_\tau(\boldsymbol{H})$ is the optimal value of the optimization problem

$$\min_{\boldsymbol{Z}} \frac{1}{2}\|\boldsymbol{H} - \boldsymbol{Z}\|_F^2 + \tau\|\boldsymbol{Z}\|_* \tag{10}$$

where $\|\cdot\|_F$ is the Frobenius norm and $\|\cdot\|_*$ shows the nuclear norm as the summation of singular values.

To perform the matrix completion part as shown in Figure 1, we recursively use the SVT in two steps. In the first step, starting with the initial matrix $\boldsymbol{Y}^0 = \boldsymbol{R}$, the singular value shrinkage operator is computed as

$$\boldsymbol{X}^k = D_\tau(\boldsymbol{Y}^{k-1}). \tag{11}$$

Then, in the second step, the difference between the projected matrix $\boldsymbol{X}^k$ and the initial matrix is compensated for the known entries using

$$\boldsymbol{Y}^k = \boldsymbol{Y}^{k-1} + \delta\mathcal{P}_\Omega(\boldsymbol{R} - \boldsymbol{X}^k) \tag{12}$$

for $k = 1, 2, \ldots$, where $\mathcal{P}_\Omega(\cdot)$ is an operator which keeps those entries of a matrix corresponding to Ω unchanged, and sets the other entries to zero. The iterations continue until the condition $\|\mathcal{P}_\Omega(\boldsymbol{X}^k - \boldsymbol{R})\|_F < \epsilon\|\boldsymbol{R}\|_F$ is satisfied and the last $\boldsymbol{X}^k$ is reported as the completed matrix.

For the purpose of haplotype assembly, we need to find $\boldsymbol{H}$ by quantizing the entries of the completed matrix of SVT to 1 and -1 as shown in Figure 1. Then, the two different rows of $\boldsymbol{H}$ are considered as the paternal and maternal haplotypes. The procedures of the HapSVT algorithm is depicted in Algorithm 1.

Algorithm 1: Haplotype assembly using SVT (HapSVT).

input : N aligned reads
output: Haplotypes $\hat{\boldsymbol{h}}_m, \hat{\boldsymbol{h}}_p$
`/* preparing the read matrix */`
Convert the sequences of nucleotides (reads) to the sequences of numbers
Add zeros to each read to construct $\boldsymbol{r}_i$s with the length of l
Construct the read matrix $\boldsymbol{R}$ $(N \times l)$
`/* SVT algorithm */`
Initialization $Y^0 = \boldsymbol{R}$, $k = 0$, $i = 1$
while $\|\mathcal{P}_\Omega(\boldsymbol{X}^k - \boldsymbol{R})\|_F < \epsilon\|\boldsymbol{R}\|_F$ **do**
  $k = k + 1$
  $\boldsymbol{X}^k = D_\tau(\boldsymbol{Y}^{k-1})$
  $\boldsymbol{Y}^k = \boldsymbol{Y}^{k-1} + \delta\mathcal{P}_\Omega(\boldsymbol{R} - \boldsymbol{X}^k)$
end
$\boldsymbol{H} = \boldsymbol{X}^k$
`/* Quantization part */`
$\boldsymbol{H} = 2 * (\boldsymbol{H} > 0) - 1$
`/* Extracting two different rows */`
$\hat{\boldsymbol{h}}_p = \boldsymbol{H}(1, :)$
while $\boldsymbol{H}(1, :) = \boldsymbol{H}(i, :)$ **do**
  $i = i + 1$
end
$\hat{\boldsymbol{h}}_m = \boldsymbol{H}(i, :)$
Convert the entries of $\hat{\boldsymbol{h}}_m$ and $\hat{\boldsymbol{h}}_p$ to the nucleotides.

4.2 Haplotype assembly using HapNuc

A popular method for matrix completion is based on relaxing the rank function to a convex function. Since the matrix rank is the number of nonzero singular values, an approximation of the rank function is given by the summation of the singular values, *i.e.*, the nuclear norm. In this way, this problem is defined as [11]

$$\min_{\boldsymbol{H}} \|\boldsymbol{H}\|_* \quad \text{subject to } \boldsymbol{H}_{ij} = \boldsymbol{R}_{ij} \text{ for } (i,j) \in \Omega. \tag{13}$$

This problem can be solved easily using the CVX, a MATLAB based package [14]. It has been shown that nuclear norm minimization has strong mathematical guarantees to achieve the optimal solution [11, 15, 16]. To develop the second new HapNuc algorithm, we substitute the SVT part of Algorithm 1 by the nuclear norm minimization.

4.3 Haplotype assembly using HapOPT

Another approach for the matrix completion part is known as the OPTSPACE [17] in which unlike the two previous techniques, for the matrix completion part we assume that the rank of the desired

matrix $\boldsymbol{H}$ is known. The OPTSPACE algorithm consists of three steps including a) trimming, b) projection, and c) cleaning as explained below.

a) In the trimming step, those columns of $\boldsymbol{R}$ with degrees larger than $2|\Omega|/l$ are set to zero where $|\cdot|$ shows the cardinality of a set and l is the haplotype length. The degree of a column or a row is defined as the number of its known entries. This step is also performed for the rows of $\boldsymbol{R}$ with degrees larger than $2|\Omega|/N$ where N is the number of reads.

b) The trimmed $\boldsymbol{R}$ obtained from Step (a) is projected to the space of rank r matrices using

$$P(\boldsymbol{R}) = \frac{Nl}{|\Omega|}\boldsymbol{U}P_r(\boldsymbol{\Sigma})\boldsymbol{V}^H \tag{14}$$

where $P_r(\boldsymbol{\Sigma}) = \text{diag}(\sigma_1, ...\sigma_r)$ and $\boldsymbol{U}$ and $\boldsymbol{V}$ are given by (7).

c) The cleaning step is performed by solving the following optimization problem,

$$\min_{\boldsymbol{X}\in\mathbb{R}^{N\times r},\boldsymbol{Y}\in\mathbb{R}^{l\times r}} \min_{\boldsymbol{S}\in\mathbb{R}^{r\times r}} \sum_{(i,j)\in\Omega} \big(\boldsymbol{R}_{ij} - (\boldsymbol{X}\boldsymbol{S}\boldsymbol{Y}^H)_{ij}\big)^2 \tag{15}$$

which contains two minimization parts. The inner part results in a function in terms of $\boldsymbol{X}$ and $\boldsymbol{Y}$. To solve the outer minimization part, we use a gradient based recursive method whose initial matrices are computed from Step (b), *i.e.*, $\boldsymbol{X}_0 = \boldsymbol{U}$ and $\boldsymbol{Y}_0 = \boldsymbol{V}$. Then, this recursive method leads to the optimal solution as $\boldsymbol{X}_{\text{opt}}\boldsymbol{S}_{\text{opt}}\boldsymbol{Y}_{\text{opt}}^H$.

To finalize the third new HapOPT algorithm, we should substitute the SVT part of Algorithm 1 by the above three steps.

5 RESULTS

Using extensive simulations, we compare the performance of the proposed HapSVT, HapNuc, and HapOPT algorithms with that of SDhaP [8]. Simulations are performed for the haplotype data and NGS paired end reads addressed in [18]. The desired haplotypes are with the length of 100. The two metrics used for comparing the algorithms are the Mean Square Error (MSE) and the reconstruction rate (rr) defined as

$$\text{MSE} = \min\big\{\|\hat{\boldsymbol{h}}_m - \boldsymbol{h}_m\|_2, \|\hat{\boldsymbol{h}}_m - \boldsymbol{h}_p\|_2\big\} \tag{16}$$

$$\text{rr} = \frac{1}{l}\min\Big\{\mathcal{HD}\big(\hat{\boldsymbol{h}}_m, \boldsymbol{h}_m\big), \mathcal{HD}\big(\hat{\boldsymbol{h}}_p, \boldsymbol{h}_p\big)\Big\}, \tag{17}$$

where $\mathcal{HD}(\cdot,\cdot)$ is the augmented hamming distance between two vectors which counts the number of sites with identical values calculated as

$$\mathcal{HD}(\boldsymbol{a}, \boldsymbol{b}) = \sum_{j=1}^{l} \mathcal{D}\big(\boldsymbol{a}(j), \boldsymbol{b}(j)\big) \tag{18}$$

where $\mathcal{D}(\cdot,\cdot)$ is

$$\mathcal{D}(a,b) = \begin{cases} 0 & a = b \\ 1 & \text{otherwise.} \end{cases} \tag{19}$$

Figure 2 compares the MSEs for different number of reads for which the corresponding reconstruction rates are depicted in Figure 3.

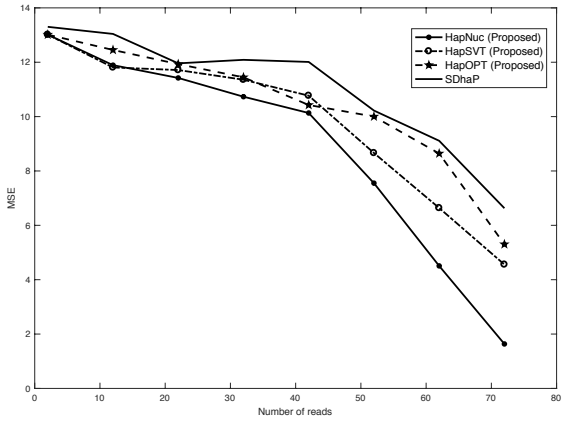

Fig. 2: Mean square error vs. the number of reads for different algorithms.

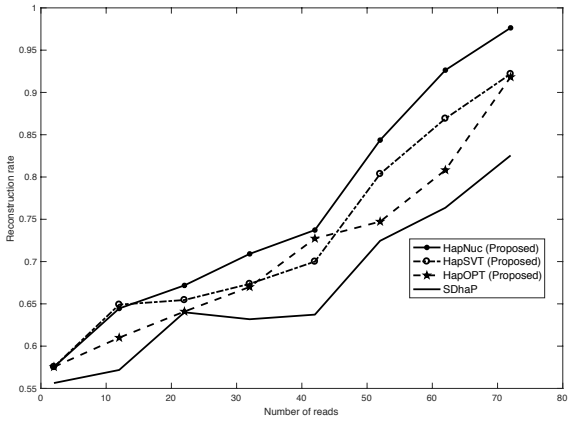

Fig. 3: Reconstruction rate vs. the number of reads for different algorithms.

As seen in both figures, the proposed matrix completion-based algorithms outperform the SDhaP. Also, the HapNuc outperforms the other ones due to reformulating the problem in convex form with a nuclear norm definition which has been deeply investigated in [11]. It is worth reminding that the SDhaP solves a non-convex optimization problem using a heuristic technique with the gradient descent algorithm which does not guarantee reaching the global optimum. Furthermore, as a consequence of increasing the number of reads, a better performance results by achieving a lower MSE and a higher reconstruction rate.

The running time of the different algorithms is shown in Table 3. As observed, the HapOPT is faster than the other ones because OPTSPACE needs just an SVD and a quadratic optimization using a gradient-based method. Although the HapNuc needs more running time in CVX, it performs better than the others in terms of the MSE and the reconstruction rate.

Tab. 3: Simulation time of the algorithms

Algorithm	50 reads	80 reads
HapOPT(Proposed)	0.0566	0.0292
SDhaP	0.0598	0.0452
HapSVT(Proposed)	0.1088	0.1562
HapNuc (Proposed)	1.7602	2.3993

Furthermore, the box plot of MSE and reconstruction rate over all experiments are depicted for each method in Figures 4 and 5, respectively. One can see that all the proposed algorithms generate lower variances compared to the SDhaP.

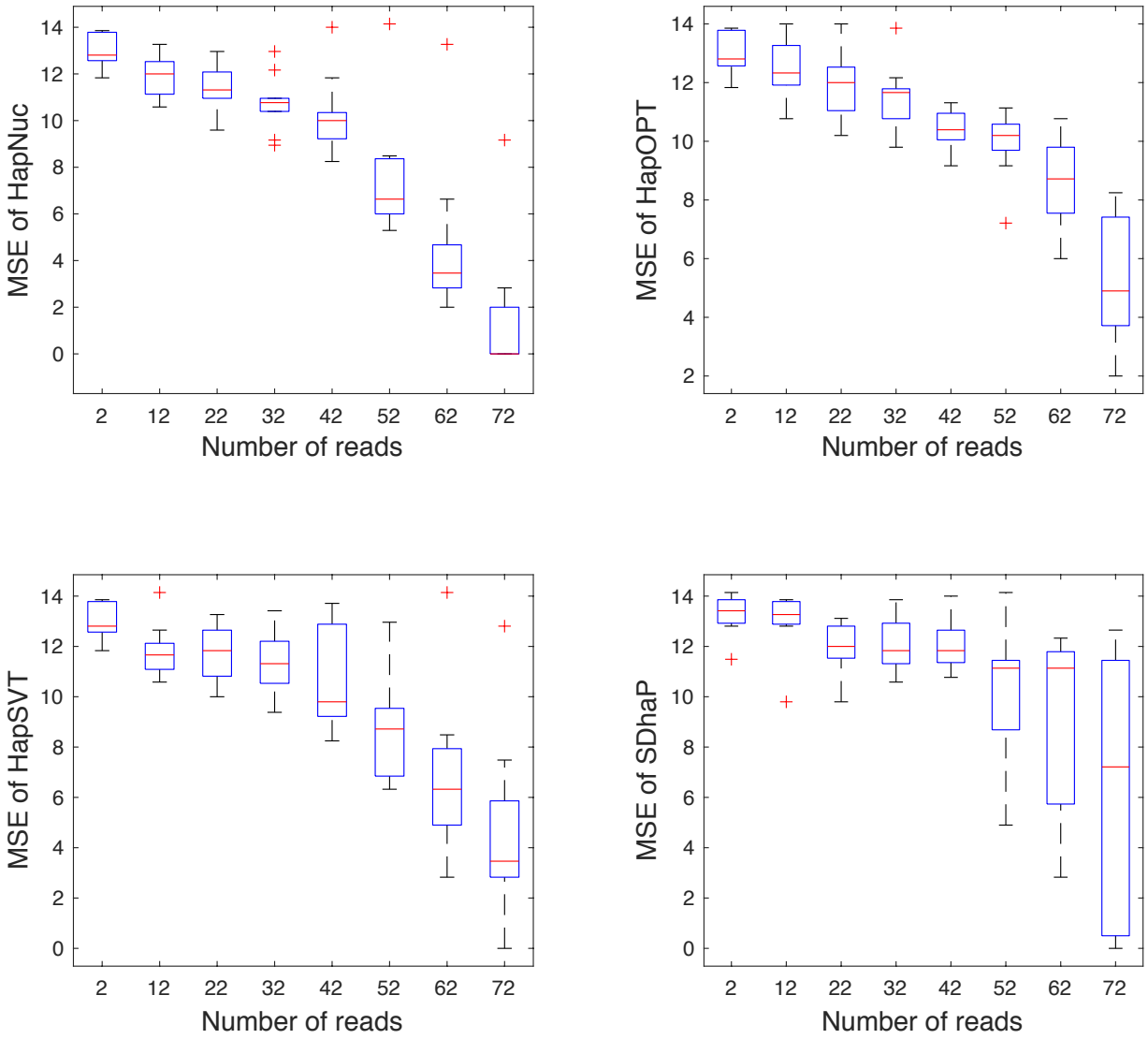

Fig. 4: Box plot of MSEs.

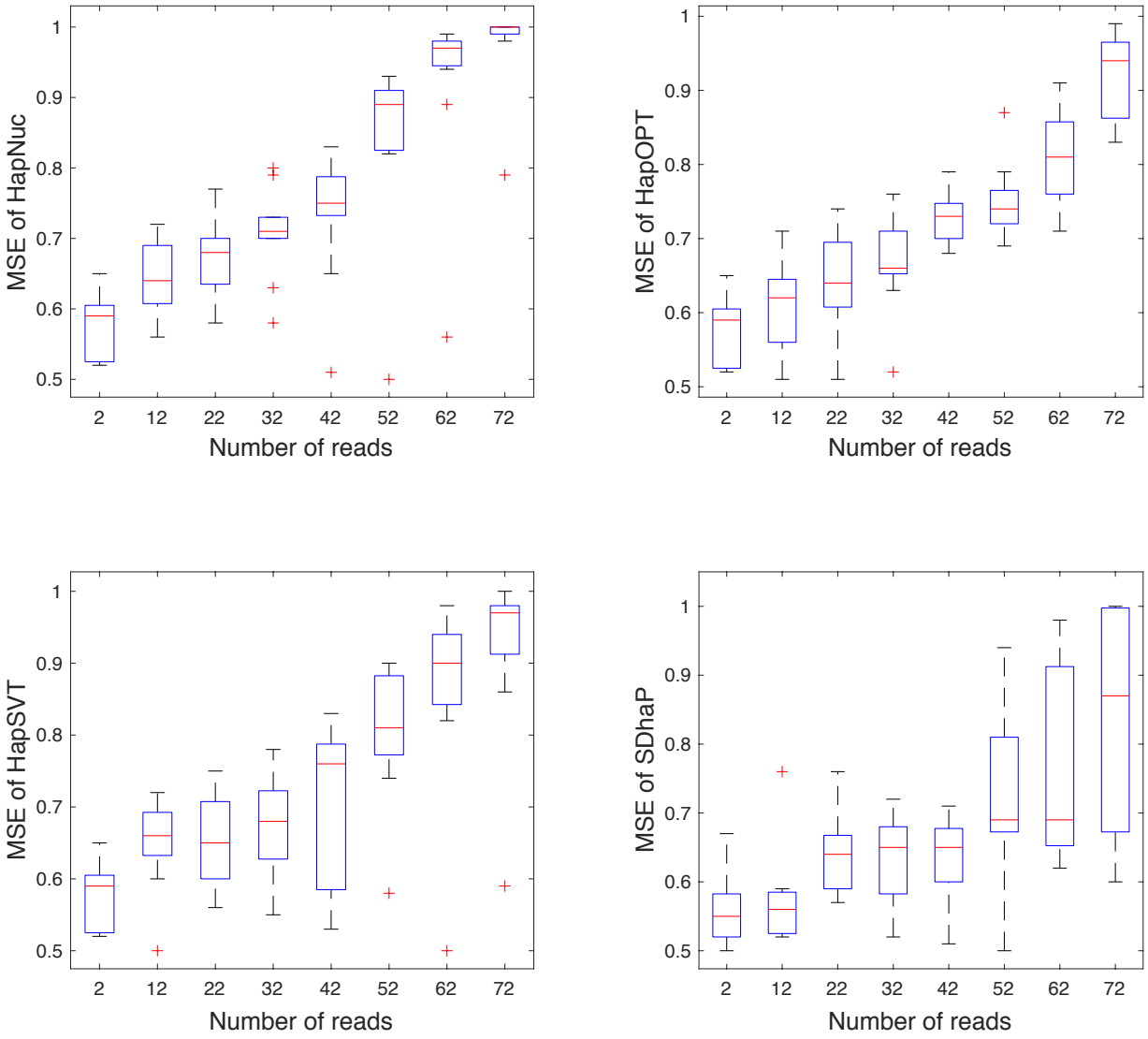

Fig. 5: Box plot of reconstruction rates.

Moreover, we evaluate the proposed algorithms for the fosmid sequence data of 22nd chromosome of the individual NA12878 with the length of 1000 analyzed in [9]. Since for the real data no haplotype exists for comparison purposes, the Minimum Error Correction (MEC) score [19] is considered as another metric which shows the fidelity of the extracted haplotype to the reads. It is assumed that the lower the MEC score, the better the quality of the haplotype extraction [9]. The normalized MEC score is given by

$$\mathrm{MEC}_{\mathrm{Score}} = \frac{1}{Nl}\sum_{i=1}^{N}\min\big(\mathrm{hd}(\boldsymbol{r}_i,\boldsymbol{h}_m),\mathrm{hd}(\boldsymbol{r}_i,\boldsymbol{h}_p)\big), \tag{20}$$

in which $\mathrm{hd}(\boldsymbol{r}_i,\boldsymbol{h}_p)$ is the hamming distance between $\boldsymbol{a}$ and $\boldsymbol{b}$ defined as [8]

$$\mathrm{hd}(\boldsymbol{a},\boldsymbol{b}) = \sum_{j=1}^{l} d\big(\boldsymbol{a}(j),\boldsymbol{b}(j)\big), \tag{21}$$

where

$$d(a,b) = \begin{cases} 1 & a \neq 0 \ \& \ b \neq 0 \ \& \ a \neq b \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

This metric for the fosmid data and the corresponding running time are shown in Table 4. As seen, among the proposed algorithms, the MEC score for the HapOPT is higher than that of the SDhaP. From the above MSE and MEC results, we can conclude that in total the HapOPT shows more accurate estimates of haplotypes with almost the same running time.

Tab. 4: MEC for the Fosmid data.

Algorithm	HapNuc(Proposed)	HapSVT(Proposed)	HapOPT(Proposed)	SDhaP
MEC score	0.002683	0.002694	0.000726	0.001050
Time (sec)	58.04	67.41	2.21	1.14

6 CONCLUSION

We have exploited matrix completion methods like the SVT, nuclear norm minimization, and OPTSPACE to estimate haplotypes more accurately. This was led to introducing the new HapNuc, HapOPT, and HapSVT algorithms. It was shown that the MSE and reconstruction rate of these algorithms outperform the recently addressed SDhaP algorithm. From the MEC score aspect, the HapOPT generated better results compared to the SDhaP for the real fosmid data with almost the same running time.

References

[1] Olivier Delaneau, Jonathan Marchini, and Jean-Fran ccois Zagury. A linear complexity phasing method for thousands of genomes. *Nature methods*, 9(2):179–181, 2012.

[2] Matthew W Snyder, Andrew Adey, Jacob O Kitzman, and Jay Shendure. Haplotype-resolved genome sequencing: experimental methods and applications. *Nature Reviews. Genetics*, 16(6):344, 2015.

[3] Lusheng Wang and Ying Xu. Haplotype inference by maximum parsimony. *Bioinformatics*, 19(14):1773–1780, 2003.

[4] Jared O'Connell, Kevin Sharp, Nick Shrine, Louise Wain, Ian Hall, Martin Tobin, Jean-Francois Zagury, Olivier Delaneau, and Jonathan Marchini. Haplotype estimation for biobank-scale data sets. *Nature genetics*, 48(7):817–820, 2016.

[5] Vikas Bansal and Vineet Bafna. Hapcut: an efficient and accurate algorithm for the haplotype assembly problem. *Bioinformatics*, 24(16):i153–i159, 2008.

[6] Emily Berger, Deniz Yorukoglu, Jian Peng, and Bonnie Berger. Haptree: A novel bayesian framework for single individual polyplotyping using ngs data. *PLoS computational biology*, 10(3):e1003502, 2014.

[7] Sayyed R Mousavi, Ilnaz Khodadadi, Hossein Falsafain, Reza Nadimi, and Nasser Ghadiri. Maximum likelihood model based on minor allele frequencies and weighted max-sat formulation for haplotype assembly. *Journal of theoretical biology*, 350:49–56, 2014.

[8] Shreepriya Das and Haris Vikalo. Sdhap: haplotype assembly for diploids and polyploids via semi-definite programming. *BMC genomics*, 16(1):260, 2015.

[9] Jorge Duitama, Gayle K McEwen, Thomas Huebsch, Stefanie Palczewski, Sabrina Schulz, Kevin Verstrepen, Eun-Kyung Suk, and Margret R Hoehe. Fosmid-based whole genome haplotyping of a hapmap trio child: evaluation of single individual haplotyping techniques. *Nucleic acids research*, 40(5):2041–2053, 2011.

[10] Changxiao Cai, Sujay Sanghavi, and Haris Vikalo. Structured low-rank matrix factorization for haplotype assembly. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):647–657, 2016.

[11] Emmanuel J Candes and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[12] Mark A Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622, 2016.

[13] Jian-Feng Cai, Emmanuel J Candes, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[14] Michael Grant, Stephen Boyd, and Yinyu Ye. Cvx: Matlab software for disciplined convex programming, 2008.

[15] Emmanuel J Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[16] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[17] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

[18] Filippo Geraci. A comparison of several algorithms for the single individual snp haplotyping reconstruction problem. *Bioinformatics*, 26(18):2217–2225, 2010.

[19] Xin-Shun Xu and Ying-Xin Li. Semi-supervised clustering algorithm for haplotype assembly problem based on mec model. *International journal of data mining and bioinformatics*, 6(4):429–446, 2012.